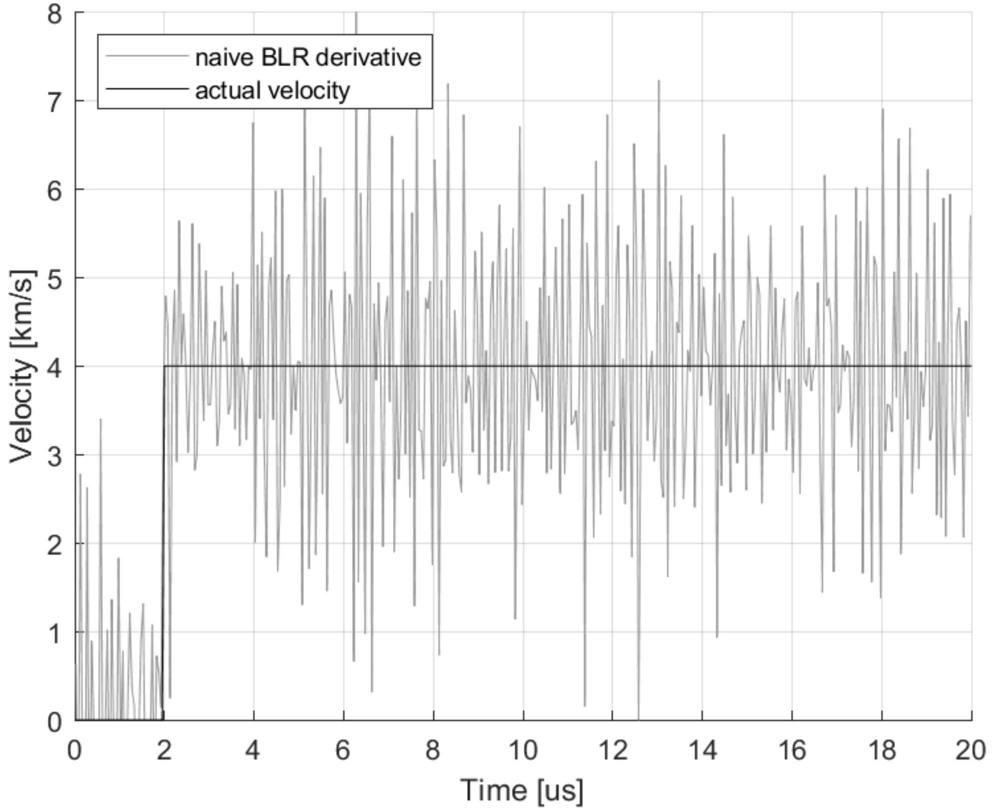
Kalman filters for derivatives of noisy data

PDV Workshop 2024

Shelly Rhodes
LLNL

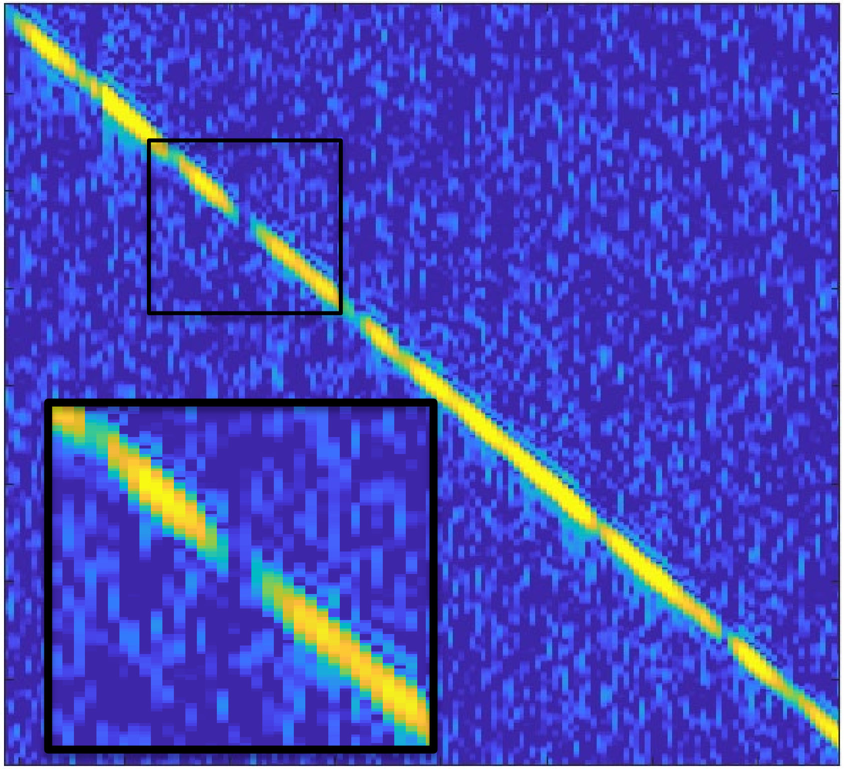
October 16, 2024

BLR extractions frequently give very noisy estimates of velocity.



Even with 100 micron position accuracy, a measurement every 50 ns allows up to 4 km/s of variation in the naïve derivative.

Light scattering from a rough surface sometimes produces poor returns.



Because BLR data is **discrete**, not continuous, temporary drops in surface reflectivity affect the data more than in continuous measurements like PDV.

We have two <u>competing</u> goals for our velocity estimation routine.

- We want to smooth the velocity, particularly where we have low confidence data points
- We want to accurately follow sharp changes in the velocity due to shocks

These are the competitors that Kalman filters have left in the dust:

- **Total Variation Regularization**
  - Optimization that balances matching data with minimizing total acceleration
  - Requires tinkering to accept variable uncertainty in the data points
  - Said tinkering makes it extremely sensitive to parameter choice
  - Good at following shocks, but tends to miss small accelerations and give velocity estimates that are too boxy
- **Local Linear Regression**
  - Basically a moving average except it can handle missing points
  - Good at following small accelerations, but smears out shocks
  - Hard to find a good compromise between reducing noise and following shocks

A Kalman filter is a matrix equation relating one or several estimated state variables to data.

- Implementation is specific to the type of system you're trying to model. Ideal for first order linear differential equations.
- Frequently used for real-time guidance, navigation, and control of aircraft/spacecraft/boats/robots/etc. System model consists of kinematic equations and control input (e.g., engine thrust).
- Only the last prediction and a new measurement are required to update the estimates.

Uncertainty is considered in every step of the calculation.

- The new estimated state is the uncertainty-weighted average of the predicted state and the new measured data point.

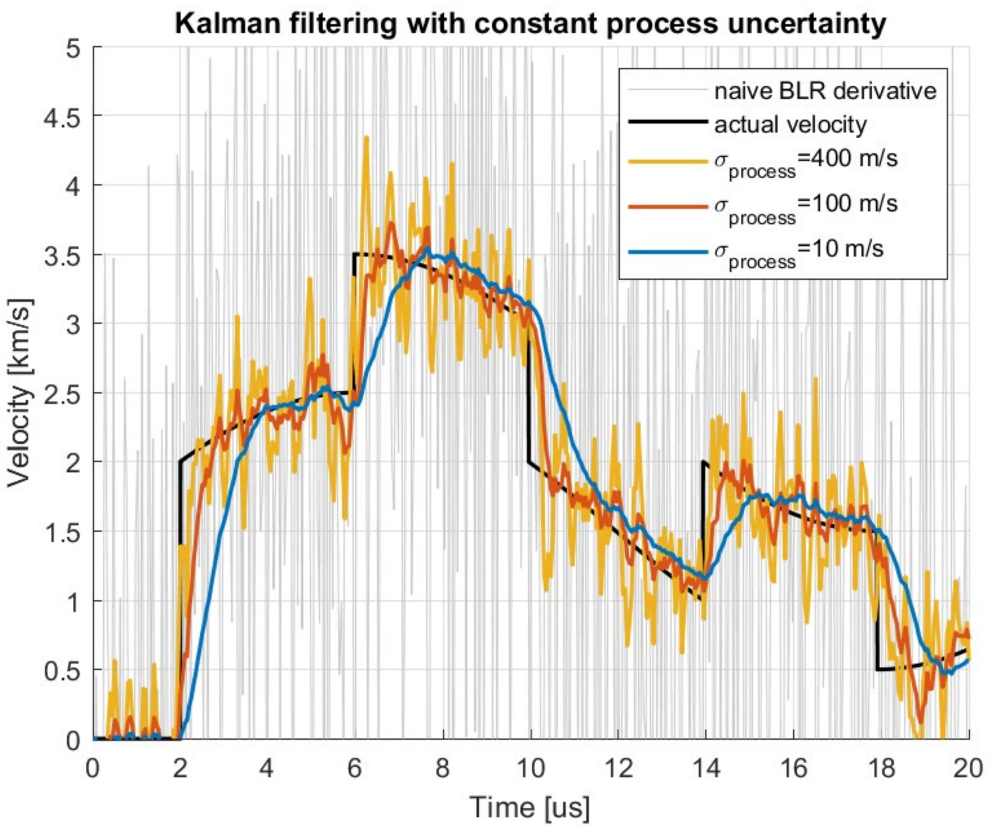
$$K = \frac{prediction\ variance}{prediction\ variance + measurement\ variance}$$

$$Estimate\ = K * (measurement) + (1 - K) * prediction$$

- When measurement uncertainty is high, the "Kalman gain" K is small, and the estimates are more highly weighted towards the predictions.

- The prediction variance is the uncertainty in the previous estimate plus the modeling (or "process") uncertainty.

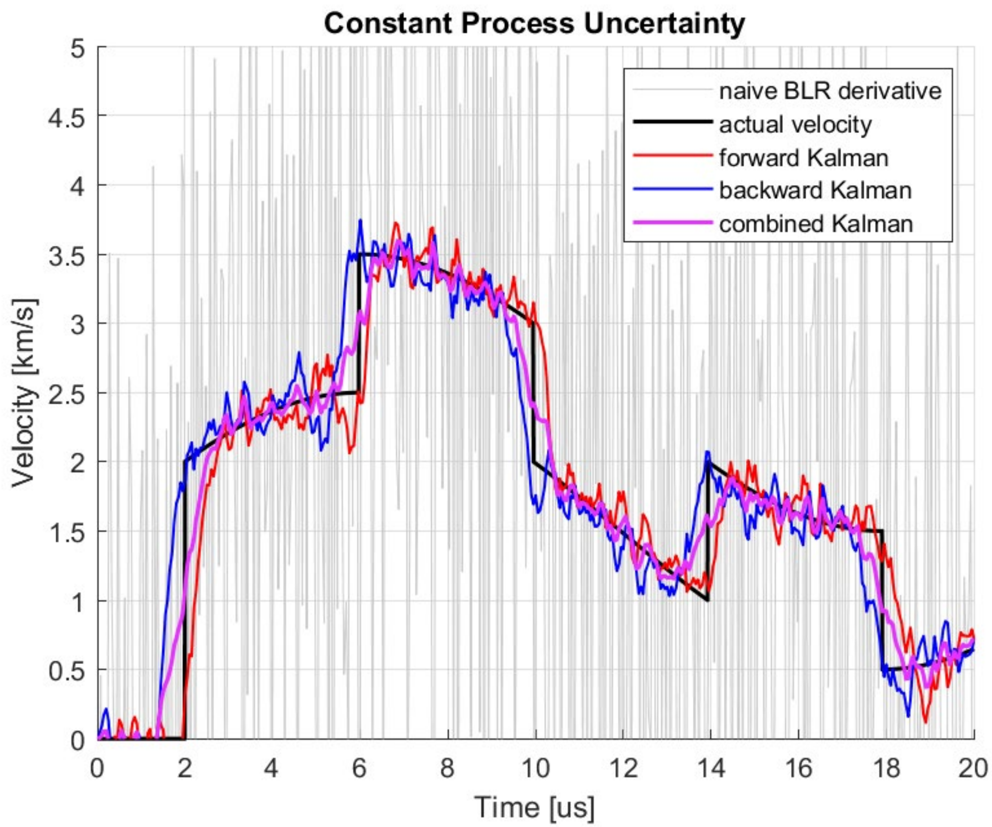
The process uncertainty controls how quickly the state variables are allowed to change.



Position curve is integrated from the actual velocity curve shown in black, then 100-micron standard deviation Gaussian noise is added before shoving it into the Kalman filter.

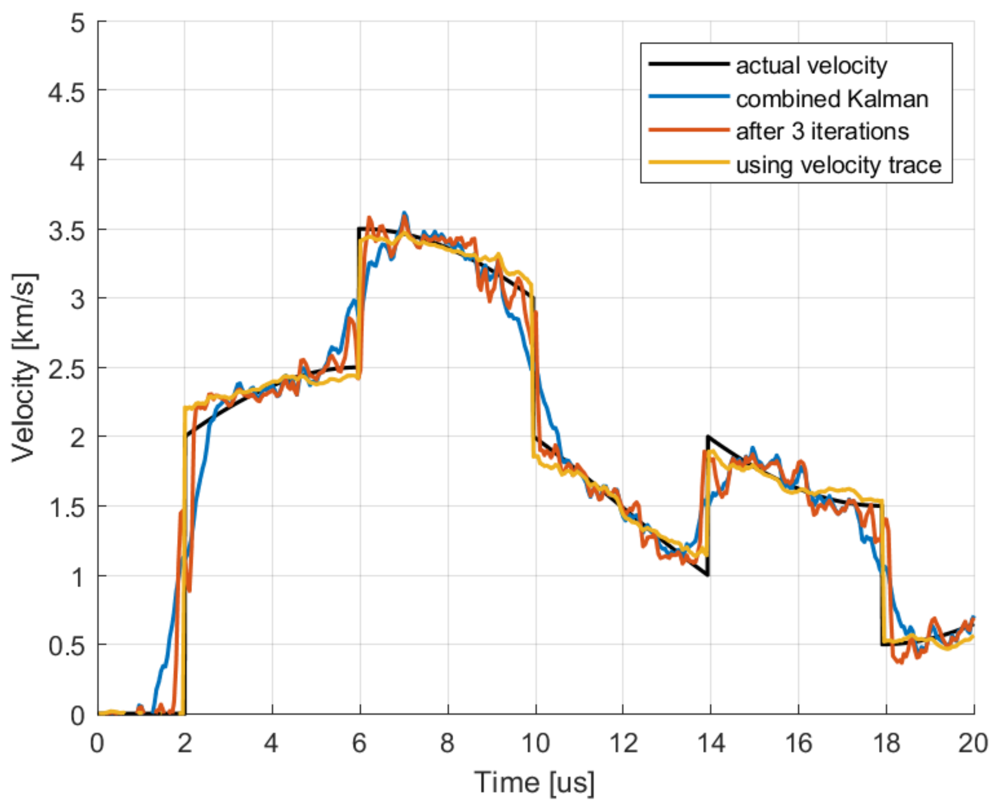
- State variables are: position and velocity. Velocity is constant with some uncertainty.
- A larger process uncertainty produces less smoothing but better responsiveness to shocks.

Lag can be fixed by running the filter forwards and backwards and combining the results.



- The shock is smeared out, but at least the ramp is centered on the actual shock time.

Can we improve velocity estimates by increasing process uncertainty around shocks?



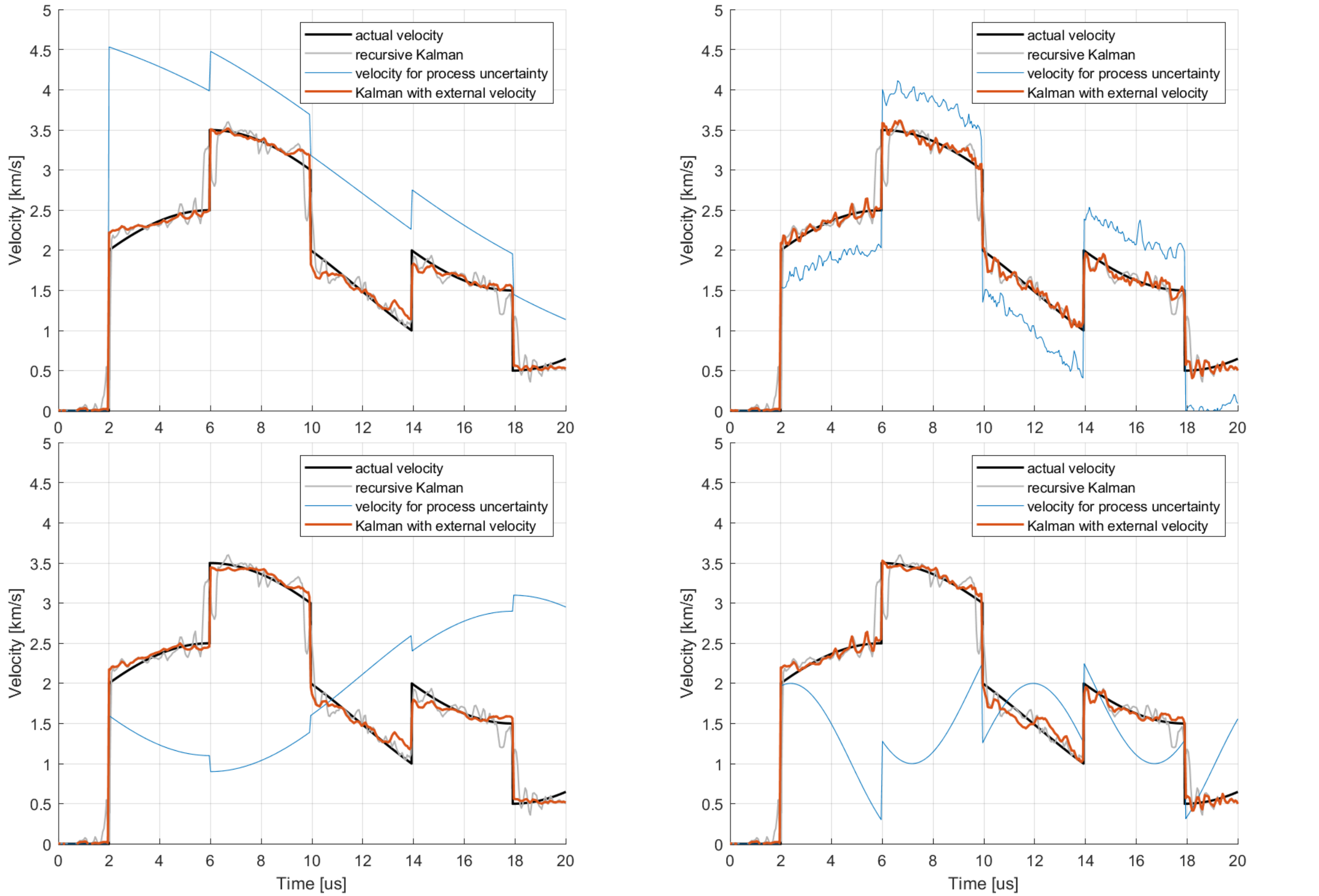
I typically use the average of the forwards and backwards velocity difference as the process uncertainty.

- Using the local variation in velocity as the process uncertainty creates sharper features around shocks.
- A velocity curve from another diagnostic is best, but a feeding Kalman-filter velocity estimate back in also improves the behavior around shocks.

Is it cheating to use a velocity curve from PDV to estimate the derivative of BLR data?

- I have thought about this a LOT
- I don't think so
- The velocity values are not used, only the local change in velocity
- The most important thing is that it increases the process uncertainty around shocks
- The input velocity curve can be wrong as long as it has shocks in sort of the right places

I'm not kidding, the input velocity curve can be very wrong and the filter still performs well.



The Kalman filter implementation I use for BLR derivatives has these important features:

- Constant velocity model (one of the simplest models)
- Accepts variable uncertainty for each position point
- Filter is run forwards and backwards and the output is the uncertainty-weighted average of the forwards and backwards results
- Option to run with a constant process/velocity uncertainty
- Option to get process/velocity uncertainty from an input velocity curve

Why a constant-velocity model? Why not include acceleration?

- At least for shock experiments, constant acceleration is not a good model for the data. Most of the acceleration is in short impulses.
  - (This also means that estimating acceleration from velocity data is often harder than estimating velocity from position data.)
- Models with acceleration tend to lag and then overshoot (as the acceleration peaks and then drops back to zero) whereas a constant velocity model only lags.
- If the acceleration in your data is gentler, including it in your filter model might give you better results!

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC